

**REMARKS**

This amendment is submitted with the Request for Continued Examination (RCE) under 37 CFR 1.114 for the subject application for the purpose of placing the application in condition for allowance. Claims 1-22 are active in the application.

5 Applicants have amended the claims previously submitted under 37 CFR 1.116(b) for the purpose of expressly reciting limitations considered to be inherent in such claims. Applicants submit that with these changes, the subject claims can not in any way be interpreted to read on prior art systems that transfer data via FTP. Even in the absence of such changes to the claims, Applicants submit that the claims should be deemed  
10 patentable under 35 USC § 102(b) over the cited teachings of Prust even when applied in the manner set forth in the Examiner's Advisory Action mailed on July 13, 2005 for the reasons discussed herein.

**The Examiner Still Has Not Established a Prima Facie Case of Anticipation**

The Examiner is required to show that each and every element as set forth in the  
15 claim is found in the Prust patent. Also, the Examiner must show that the identical invention is shown in as complete detail as contained in the claim and that the elements must be arranged as required by the claim.

The Examiner cited as a basis of the Final Rejection of Applicants claims the technique for transferring data in between is by using FTP or browser and also  
20 Applicants' specification at paragraph #7, wherein FTP is characterized as a technique for transfer of files between heterogeneous computers in the prior art. Also, the Examiner cites paragraph #7 of Applicants specification relative to step (b) of Applicants' claim 1. Applicants submitted that the grounds of rejection was in **essence a new rejection based** on the description pertaining to transferring data by FTP contained  
25 in paragraph #7 of Applicants specification.

As discussed in Applicants prior responses, Prust is directed to a data storage system (e.g. storage servers) that provides **several methods of accessing virtual storage areas** by client computers. In one embodiment, access to the virtual storage areas is from a user directly via the operating system's user interface by calling standard file  
30 management routines provided by an API of the operating system (e.g. for copying files between hard disk and remote storage areas as well as renaming files and deleting files)

(referred to herein as method 1). Prust states that in the method 1 embodiment, the client operating system of the client computers is the Macintosh operating system, such that the API includes the Apple File Services (AFS) and storage servers support accessing remote data files within virtual storage areas via the Apple Filing Protocol (AFP) services over TCP/IP. Prust also discusses another embodiment of Figure 4 in which the operating system is a Windows operating system from Microsoft that incorporates the SMB protocol or WebDAV protocol. Here, it will be noted that the file services on the client and server systems are described by Prust as **being compatible or homogeneous**. Applicants direct the Examiner's attention to paragraphs #10-12 of Applicants specification which also discuss the use of functionality of writing files on remote systems by client computers prior art in **homogeneous systems** using the Sun Solaris operating system or Microsoft Windows operating system.

In another embodiment of Prust (herein cited as method 2), the user can access virtual storage areas by invoking a communications application such as a web browser or an FTP utility. In this embodiment, the communications application typically uses TCP/IP as the base protocol and additionally uses the HTTP protocol (used for transferring documents/text), the FTP protocol or even a proprietary data backup protocol. In this embodiment, the communications software application handles all communications with the storage servers and Prust specifically states that **the file management routines of the operating system (i.e. API) are not invoked** (see lines 53-54 of column 6 of Prust). Thus, Prust treats these two methods of access as being distinctly different. This difference is also specifically recited in claim 1 of Prust and is also claimed in related patent no. 6,735,623 to Prust entitled "Method and system for accessing a remote storage area issued on May 11, 2004 (hereinafter Prust I). The claims of this patent are directed to the different access interfaces included in data storage system of the cited Prust patent. Claim 18 of the Prust I patent recites the following: "The data storage system of claim 10 wherein the plurality of access interfaces further comprise a third access interface to service access requests from at least one communication software application executing on the client computer **and that bypasses the API of the operating system that present the target one of the plurality of remote storage areas to software applications executing on the client computer as local to**

the client computer". A further claim (claim 21) of the same patent recites a data storage system that provides a plurality of different access interfaces for accessing remote storage areas wherein the access interfaces comprise a first access interface wherein the API supports Web Distributed Authoring and Versioning (WebDAV) for accessing files within the target one of the plurality of user-assignable storage area using extensions to the Hypertext Transfer Protocol (HUP) . . . , a second access interface to directly service access requests from at least one communications software application executing on the client computer to automatically backup files from the client computer to the data storage system and a third access interface to allow a web browser executing on the client computer to browse the target one of the plurality of remote storage areas. Both Prust patents make it clear that the different access interfaces are to be treated as separate and distinct from one another. Any attempt to combine the different access interfaces would be contrary to the teachings of Prust and further would render the resulting combination inoperative.

As discussed in Applicants prior responses, in rejecting claim 1, the Examiner makes reference to portions of Prust that pertain to the above discussed two conceptually different embodiments of access methods #1 and #2. More specifically, relative to steps (A) and (B), the Examiner cited the embodiment of method #1 which uses the file management routines of the operating system API and relative to step (E), the Examiner cited the embodiment of method #2 (does not use the file management routines of the operating system (API)) and specifically cites the FTP protocol as one of the types of communications software. Applicants submitted that clearly, it was improper to combine the elements/steps of what Prust discloses as two different methods of accessing virtual storage areas described as not to be combined in order to meet the requirements of 35 U.S.C. 102 as the Examiner has done. If this were not true, then one could pick and choose among different disclosures in a patent and combine them in any way and contrary to the teachings of the patent that would arguably anticipate the claims in question.

Also, Applicants noted that there are no specific portions of Prust or any other reference cited relative to steps (C) and (D) of claim 1. Hence, one would conclude that these steps are not anticipated and hence, the case for anticipation has not been shown.

In response to this argument, the Examiner in the Advisory Action mailed July 13, 2005, stated that Prust teaches that a set of API routines are both required by access methods #1 and #2 (the latter being the FTP method citing column 6, lines 44-67. The Examiner further noted in the Advisory Action that when Prust uses Macintosh's Apple File Services as an example of access method #1, Prust teaches that "API includes the Apple File Services (AFS), ...via the Apple Filing Protocol (AFP) services over TCP/IP (column 6, lines 6-12). The Examiner based on the cited description concluded that it was clear that such teaching did not exclude FTP from using API routines as local file managing routines but that Prust taught including AFS as part of the API routines and replacing FTP with the AFP protocol for access method #1.

From the above, Applicants now understand that the basis for rejecting claim 1 is that the claims are anticipated by access method #1 when modified by the substitution of the FTP protocol for AFS protocol. First, Applicants submit that the basis of Examiner's rejection comes under the provisions of 35 U.S.C. 103 and not the original grounds of rejection under 35 U.S.C. 102(b). Second, such substitution is contrary to the previous cited teachings of the Prust patent. Accordingly, claim 1 should be deemed patentable over the teachings of the Prust patent.

**Amendments to Applicants claims**

In response to the Examiner's statement that Applicants have not clearly defined the word "heterogeneous", Applicants have amended the claims to specify that the term "heterogeneous" includes systems having different file formats and word structures consistent with the use of the term in the prior art. Also, Applicants have amended the claims to expressly state that the plurality of blocks being transferred constitute only a portion of the file being accessed and not the complete file as required by FTP methods such as described relative to Figure 2 of Applicants specification and discussed in prior responses.

It will further be noted that the claims require that the blocking of plurality of records into plurality of blocks is accomplished via the API used to communicate with the first interface. Clearly, none of the prior art references show or suggest the use of such an API. The Examiner stated in the grounds for rejection that the cited "packetizing data files" reads on the claims because each packet may be equivalent to a block. As

discussed in Applicants prior response, the term “packetizes” refers to the process of converting the data files and metadata into “packets” for transfer over the network which occurs not at the application level. As stated above, the description in Prust makes it clear that the user is accessing complete files and performing file management operations such as copying, renaming, moving and deleting files and directories (see column 6, lines 1-12). It should be obvious that if the files in question are sufficiently large relative to the size of a packet, then it would become necessary to break down the data in the files into a number of packets for transmission over the network. Thus, the process of “packetizing data files” is opposite to the steps recited in Applicants claims. Packetizing is further discussed herein.

Additionally, the Examiner cited paragraph #42 of Applicants specification in support of the Examiner’s opinion that Applicants admit that “data transfers between systems is typically on a (blocked) record basis” in prior art systems. Applicants pointed out that a closer reading of paragraph #42 reveals that the paragraph pertains to the present invention and not to the prior art.

Paragraph #42 states the following: “Another **improvement** has been made to the prior art (**referring to the present invention**). The application 130 on the first computer system 110 can specify what data conversions are to be performed by the interface between systems. Since the data transfers between systems is typically on a (blocked) record basis, this data conversion can be selected on a per field basis, and is performed on each selected field in each record transferred. Thus, some fields can be converted automatically from 36-bit integers to 32 bit integers (and potentially reversing the “endian” for the integers at the same time), while other fields can be converted from 9-bit ASCII to 8-bit ASCII.” This latter aspect of the present invention is reflected in Applicants claims 9, 10 19 and 20. Also, in the prior art FTP method of moving data described in paragraph #7 which is discussed herein in the paragraph “Applicants Invention, an additional step would have been introduced; this step is labeled (3):

(1) Reading data from a production database and writing it to a local file; (2) utilizing FTP to transfer the local file to a remote system; (3) **Running a process on the remote system that reads the transferred data, transforms it, and writes it to another file** and (4) Running a process on the remote system that uses the transferred data. From this

discussion, it should be clear that paragraph #42 describes the present invention and not the prior art.

### **Applicants Invention**

By contrast, the present invention enhances the prior art method of moving data  
5 between two systems. Consistent with the description in paragraph #7, the method of the prior art technique includes the basic steps of: (1) reading data from a data source such as a database and writing them to a local file with some possible restrictions on some data; (2) utilizing FTP to transfer the local file to a remote system; and (3) running a process on the remote system that uses the transferred data. The present invention eliminates the  
10 need for step (2) and thus improves performance in addition to eliminating the need to integrate FTP into various operations such as for example, an unattended batch operation that can be performed overnight. At this point, it seems useful to discuss the prior art protocols cited by the Examiner in rejecting claim 1.

### **TCP/IP FTP**

15 The Transmission Control Protocol and IP are layered protocols that fit below the network-applications protocols and above access and data-link protocols. The Internet Protocol resides at about the Network Layer (Layer 3) of OSI and TCP at the Transport Layer (Layer 4). File transfer systems making use of TCP/IP typically fall at Layer 5 (Session). TCP/IP was developed as a means for tying together diverse networks with  
20 gateways. It uses checksums, sequence numbering of all data, retransmissions for reliability, reliable connection establishment and clearing and a flow-control mechanism.

The file-transfer process as it is managed by FTP is not intended to handle all issues or steps of the process just described. Rather, FTP presumes some basic properties, such as data type, file organization, and file ownership, and provides a means  
25 by which one computer can manipulate these properties on another computer system without either computer knowing details about the other. Files on a remote system are manipulated through a series of commands and responses, performing such functions as **get a file from** or **send a file to** a remote system. The File Transfer Protocol itself does not translate files from one computer type to another nor does it establish any kind of  
30 virtual network file. More fundamentally, it provides three dimensions: data types, file types and transmission modes. These dimensions can then be used by the two computers

to establish a common ground. One of the transmission modes provided by FTP is **block mode that is usually used with very large files. In this mode, the source host breaks the data into well-defined blocks and the destination machine reassembles the blocks into an appropriate file.**

5 Problems such as data extraction, formatting and translation are accomplished prior or subsequent to submission to FTP in the Application Layer of the OSI standard. A file transfer using FTP can be initiated by either a human or another computer program. The human user will use a program called the **User FTP Program**. User FTP provides access to all FTP services. In some installations, FTP may also be called (like a  
10 subroutine) from other application programs. This approach is used if there were complex data extraction, formatting and translation issues to be handled. (See pages 291-295 of the text entitled "Enterprise-Wide Computing: How to Implement and Manage LANs" by Thomas W. Madron, copyright © 1991 by Thomas W. Madron, published by John Wiley & Sons, Inc.), previous submitted in Applicants prior response.

15 Additionally, U.S. patent no. 6, 718, 372 to Bober cited by the Examiner as being pertinent to Applicants disclosure also describes a prior art technique that uses FTP for obtaining access to data stored on a remote computer system in column 3 of the patent. The description in column 3 makes it clear that FTP transfers **the entire contents** of a requested file. According to Bober, the technique uses FTP to provide a connection  
20 between an FTP server and an FTP client **to transfer an entire file**, for example, from the mainframe to a workstation. According to the patent, a user application can invoke the FTP client directly using an FTP command to cause the FTP client to request **the entire contents** of one or more files from the FTP server. In response to such an FTP command, the FTP client provides standard FTP protocol messages over the network to  
25 the FTP server. In response to such messages, the FTP server finds and then transfers the **entire contents** of the requested file(s) obtained from the storage device back to the FTP client on the workstation via the network. The FTP client receives the data during the transfer and stores the data into a file created within the local storage device on the workstation. Once the transfer is complete, the FTP session is over and the application  
30 can access the copy of the requested file as needed directly on the local storage device.

Bober points out that the FTP is generally more limited in its capabilities than

NFS since **FTP merely provides a complete local copy of an entire file.** Also, Bober does not consider FTP to be a true real time data access protocol in that the data access by an application takes place generally after the entire file has been transferred to the destination local storage device. Since FTP provides only a copy of the data file for use  
5 by the application, changes to the original file that occur after the FTP file transfer is complete may not be reflected in the copied version of the file stored within the local storage device.

From the above, it is clear that there are important distinctions between FTP file transfers and that of the present invention and from this it can be appreciated how  
10 performance can be improved by the removal of the FTP step by applying the teachings of the present invention. As set forth in claim 1, the present invention performs such moving of data by providing a record connection oriented API between the programs using it which in the preferred embodiment corresponds to GCOS 8 Cobol-85 API as described in Applicants specification (e.g. paragraphs #105, 332). The API is used by  
15 client programs in carrying out the steps of claim 1 as indicated. For example, step (A) of claim 1 would be carried out in the preferred embodiment by a program calling an **Open** function of the API that establishes a connection to the second computer (e.g. via the sockets interface which is at a lower layer of the OSI standard).

As indicated in step (B) of claim 1, the first computer blocks a first plurality of  
20 records into a first block of records via the API. In the preferred embodiment, the program calls a function **Write a Record** causing the Record Manager component to move a record into a collection buffer for the connection (see section 3.1.2 of and section 5.1.7 of Applicants specification). The first program repeatedly calls this Record Manager function resulting in the filling of the collection buffer. In the case of the  
25 present invention, since the blocking of records is done **prior to the call** to the sockets interface, there is no need to perform repeated communications with the sockets interface for each record (see section 4.1.3 of Applicants specification). This results in a substantial performance improvement.

The absence of the above discussed elements and functions should be persuasive  
30 that Prust does not anticipate Applicants claim 1. A notice to this effect is respectfully solicited. If the Examiner persists in this rejection, Applicants ask that the Examiner to

be more specific as to the portions of Prust being relied on for concluding that claim 1 is anticipated.

**Claim 2**

For the reasons set forth in Applicants prior response to the Final Rejection,  
5 Applicants submit that Prust does not provide for the receiving of the first plurality of records via the API or the writing of the second plurality of records to the first file as set forth in claim 2. Claim 2 treats a plurality of records as individual entities which are collected to form a first plurality of blocks and are unblocked into a second plurality of records for writing into the first file. This is in contrast to writing a plurality of records as  
10 a first file using a single file-oriented command. Accordingly, claim 2 should be deemed patentable over the cited portions of the Prust patent. A notice to this effect is respectfully solicited.

**Claim 3**

The Examiner commented that it is inherent that an FTP program is able to  
15 transfer files in both directions. In Prust, the objective is to enable a user to access a virtual storage area on the remote system that provides several separate and distinct data storage access interfaces as discussed above. Thus, Prust contemplates a different architecture which would have to be modified extensively in some unknown manner to accomplish bidirectional file transfers. Further, Applicants claim is not directed to an  
20 FTP transfer but rather to a method that eliminates the need for FTP transfers as discussed above.

**Claims 11 and 13 and 21-22**

In view of the above discussion, Applicants submit that claims 11 and 13 and 21-  
22 should also be deemed patentable for the same reason set forth relative to claims 1-3  
25 and 11.

**Other Rejected Claims**


Applicants response to the Final Rejection discusses in detail, reasons as to why Applicants claims 4 and 14, 5-8, 15-18, 9-10 and 19-20 distinguish patentably over the teaches of Prust and cited undocumented statements.

In view of the above arguments and claim amendments, Applicants submit that claims 1-22 should be deemed patentable over the cited prior art. A notice to this effect is respectfully solicited.

Applicants ask the Examiner to contact Applicants attorney upon receipt of this amendment for the purpose of advancing the prosecution of this application.

Further, if any questions or issues should arise with respect to this amendment or the allowability of this application, the Examiner is **urged to call Applicants' attorney at the number indicated herein.**

Respectfully submitted,



Faith F. Driscoll  
Registration No. 24,206  
Attorney for Applicants  
(781) 326-6645

FFD/fd